SERVICE ORIENTED ARCHITECTURE:

An In-Depth Look at Why IBM Software Provides a Better SOA Foundation Than Microsoft and SAP NetWeaver

IBM Software Group

Executive Summary

The evolution of information technology and rise of the Internet has unleashed new opportunities for businesses to reap the benefits of integrating their disparate business systems and processes. This opportunity comes at a time when businesses are becoming ever more complex in their interactions with partners, suppliers, and customers. Occurrences like mergers/acquisitions, restructurings, and regulatory changes are constantly changing the business processes touching each of the above entities.

Service Oriented Architecture (SOA) is widely seen as the next-generation IT infrastructure that will enable enterprises to integrate their business processes end-to-end across the company and with key partners, suppliers and customers. This will enable companies to respond with speed to any customer demand, market opportunity, or external threat.

What Is SOA?

SOA is an architectural style whose goal is to achieve loose coupling among interacting IT components. SOA-based IT infrastructures consist of applications enabled as services that are published as "available" on an enterprise service bus. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. In such an implementation, a business process can be defined as an orchestrated interaction of various service providers and consumers. This approach gives enormous IT flexibility, reduces redundancy, encourages reuse, and increases accountability.

Comparing SOA Capability

Web services and other open standards have accelerated the adoption of SOAs. Many IT vendors have introduced various tools and platforms to enable SOA implementation.

In this paper we examine the SOA capabilities of IBM, Microsoft and SAP NetWeaver products along three important criteria:
1. Developer tools to enable applications as services.
2. Service-bus middleware to facilitate mediation between services.
3. Pre-built service-combining frameworks for people integration (portal and collaboration), business integration, and information integration to jumpstart SOA adoption by enterprises.

Key Findings

IBM WebSphere Studio Application Developer provides an advanced software development platform to rapidly build new services and enable existing IT components as services. In comparison, Microsoft and SAP NetWeaver development environments suffer from the problems of multiple programming models. They lack wizards to automatically convert existing enterprise IT applications into services and hence require significant development effort.

IBM also scores with an advanced Enterprise Service Bus which has powerful mediation capabilities. WebSphere MQ (WMQ) is the industry leading messaging product and WebSphere Business Integration Message Broker and WebSphere Application Server provide high performance message routing, transformation and web services gateway services. In comparison, Microsoft and SAP NetWeaver lack a web services gateway and have limited mediation capabilities. Messaging in Microsoft is limited only to

the Windows platform and SAP NetWeaver requires a third party messaging product like WMQ for messaging outside the SAP system.

Finally, the service combining frameworks provided by IBM products are much superior to those of Microsoft and SAP NetWeaver.

WebSphere Portal has powerful built-in features for collaboration, process choreography, and content management. Microsoft SharePoint and SAP NetWeaver Enterprise Portal either are limited in providing such features or require the purchase of additional products to enable such features. In addition they do not support key JSR portal standards.

For business integration/process choreography, WebSphere provides advanced products to rapidly model, deploy and monitor business processes. Unlike WebSphere, Microsoft and SAP NetWeaver do not have simulation and analysis capabilities for process modeling. SAP NetWeaver does not have any monitoring tools and in a Microsoft environment process monitoring is not integrated with modeling. In addition, the Human Workflow Services capabilities of SAP NetWeaver and Microsoft are very limited.

WebSphere Information Integrator has advanced federation capabilities to integrate data from structured and unstructured data sources like relational databases (Oracle, SQL Server, DB2 etc), various content management repositories (Documentum, IBM Content Manager, Exchange public folders, Domino.Doc etc), file systems, news groups, text documents etc. While SAP NetWeaver does not have any federation capabilities, Microsoft's federation capabilities are limited only to standard ODBC sources.

Based on our extensive analysis, we have found that IBM has by far the most advanced SOA tools and capabilities.

What Is Service Oriented Architecture?

A Typical Problem

A major airline has just acquired a smaller regional player. The company saw an opportunity to tap into the commuter market on the East coast — Boston, New York, Washington D.C. The clientele is there. The routes are there. But so is a big problem. The company can't process the new passengers they've acquired. Why? Because the parent company's enormously complex IT systems, cobbled together over decades, have no way to communicate with those of the new acquisition. Building and testing a new system from the ground up is far too expensive, and with 90 days before the merger is complete, they simply do not have the time.

Sound familiar? The problem, of course, is an inflexible IT infrastructure. Over the years most companies have been adding and removing applications, connections, and new technologies with no long-term plan for how they could work together. The occurrence of business changes like corporate mergers, acquisitions, business restructurings, and regulatory changes has become commonplace, confronting IT departments with increasingly complex integration requirements.

Over time, IT environments become very complicated, made up of systems that cannot communicate with each other effectively, or at all. This is because every time one of those applications or connections is added or removed, the change has to be custom coded – "hardwired" – before the applications recognize one another and do what they should. The process is especially burdensome when connections involve business suppliers or other third parties, slowing time to market. Eventually, nothing can move or change without an entirely new set of instructions being written.

The Solution – Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) is an application framework that takes everyday business applications—such as SAP® Order Management, Oracle® Financials, or legacy applications—and breaks them down into individual business functions and processes, called services, that can be reused and recombined to support different activities. A service may be a business function such as *check customer credit*, or it may be a system capability such as *login user*. An SOA enables enterprises to build, deploy, and integrate these services in a flexible and agile manner, independent of the applications on which they run.

Reusing software components is not a new concept—Object Oriented Analysis and Design (OOAD) had a similar aim. While the SOA approach strongly reinforces well-established, general software architecture and OOAD principles such as information hiding, modularization, and separation of concerns, it also adds additional themes such as service choreography, service repositories, and the service bus middleware pattern [1]. Progress in multiple technologies like mediation services, service combining frameworks, and

development of open standards allows an SOA to take the concept of reuse of services to the next level [2].

Open Standards and Web Services

Until now, connections between services in an SOA had to be custom coded, which can be expensive, time consuming, and inefficient. The true value of SOAs has been limited because of a lack of open standards for integration between services. That's where Web services come in. *Web services employ open standards to maximize the value of an SOA [3].*

Web services are self-contained, modular applications. Because they are based on open standards, Web services are able to work together without relying on custom-coded connections. The services, or components, share a common protocol, or way of communicating. In other words, they can understand each other even though they don't speak the same language or share the same application platform. Web services provide rules of engagement that let service components interact dynamically as circumstances require.

With Web services, a company can make individual services available via a network to other systems. For example, the service *check customer credit* — residing on a Microsoft® Windows® server running IBM® WebSphere® Application Server in Raleigh can be invoked by the service *ship to customer* — written in Java™ code and residing on a Linux® server in Los Angeles.

Key Web service interfaces defined by standards groups include:

***Business Process Execution Language for Web Services* (BPEL4WS or BPEL**). BPEL provides a language for the formal specification of business processes and business interaction protocols [4]. By doing so, it extends the Web Services interaction model and enables it to support business transactions.

***Web Services Description Language* (WSDL).** WSDL is an XML-based language for describing Web services, including the location of the service as well as the methods the service exposes [5].
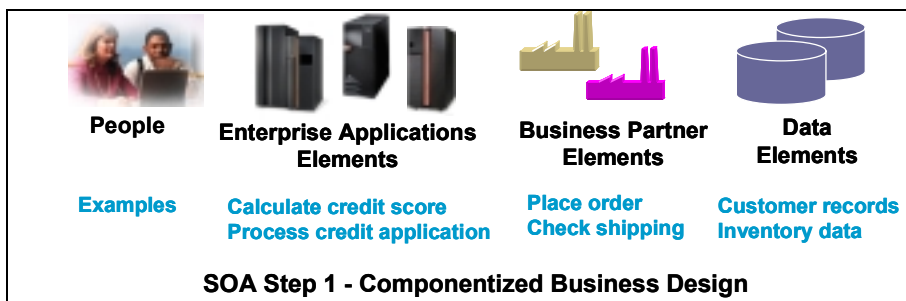
***Simple Object Access Protocol* (SOAP**). SOAP is a lightweight XML-base protocol for exchange of information in a decentralized, distributed environment [6].

Building Blocks for SOA

Start with a Componentized Business Design

The first step in building an SOA environment is to identify the various independent components that make up a business process. Each component is a logical grouping of the people, technology, and resources that deliver specific business value, with the potential to operate independently. Components have well-defined interfaces—each receives input, processes relevant tasks, and outputs the results to other components. Breaking up a business process into such components has several advantages:

1. Facilitates reuse of components: A component built for a particular business process can be reused for another business process. For example, an application called "Calculate Credit Score" is an IT component that can be used in various business processes—e.g., "Approval of Customer Application" or "Renewal of Customer Account."

2. Enables flexibility and compartmentalizes change: Because a business process is made of several components linked together, any change in the business process may be achieved by adding/removing the components to the process and/or only changing the manner in which they are linked. The components themselves may not have to be changed.

3. Reduces redundancy: If a component in one business process is similar in function to a component in another business process, perhaps both components can be merged to form a single component that can be used by both processes. Enterprises can significantly reduce costs by identifying redundant IT components and merging them—achieving lower administration, maintenance, and management costs, and potentially reduced software licensing and hardware costs.

4. Ensures accountability: Most of the business processes in today's complex businesses span several organizations and departments. Hence, it is often difficult to hold a particular organization accountable in the event of a failure in the business process or to even identify the point of failure. With a componentized SOA model, each component can be assigned to a particular department who owns it and is responsible for its maintenance.

People	Enterprise Applications Elements	Business Partner Elements	Data Elements
Examples	Calculate credit score Process credit application	Place order Check shipping	Customer records Inventory data

SOA Step 1 - Componentized Business Design

Expose Elements of Business Processes as Services

Once we have established IT components, the next step is to expose the components as services that can be consumed by requestors—by people or other IT components. The components are encapsulated as services with a service interface. The service could be created by first implementing the service and then defining the interface, or by first defining the interface and then implementing the service. In either case, the interface and at least part of the service runs in an application server environment (e.g.,

WebSphere/J2EE™ for IBM or Windows Server™ 2003/.NET for Microsoft). The application server may in turn be providing access to a database (DB2®, SQL Server, or others), enterprise information systems (CICS®, IMS®, SAP, etc.), other services, or other applications.

However, Web services are not the only way to represent IT components as services. IBM's WebSphere platform [7] enables all IT components, not just Web services, to be represented as services. IBM's more comprehensive SOA approach allows new applications to reuse resources without the development effort of first converting them to Web services. Any IT component—EJBs, Message Driven Beans (MDBs), Java classes, CICS—can be given a Web interface (WSDL) and represented as a service.
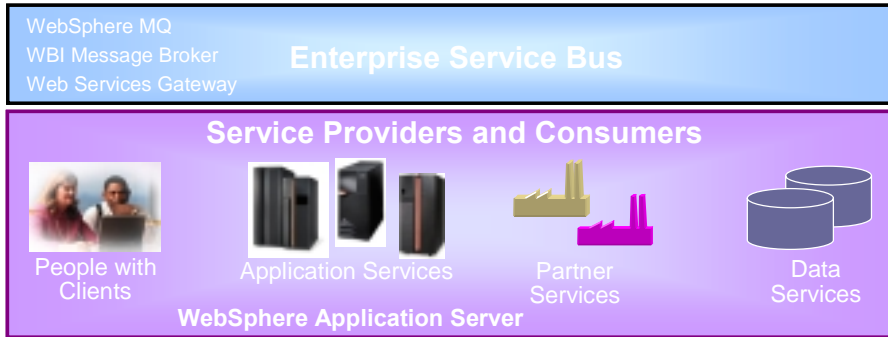

SOA Step 2 – Expose Elements of Business Processes as Services

Connect Services with an Enterprise Service Bus

The next step is to connect the various services through an Enterprise Service Bus (ESB). An ESB provides mediation services through message-based communication and message transformation and routing. A powerful ESB is essential for a robust SOA implementation. The main features of an ESB are:

1. Connection: The ESB should be able to connect various applications regardless of the platform and location. It should provide assured message delivery in a secure and reliable manner and should be able to handle messages in high volumes.

2. Message Transformation: An ESB should be able to transform messages from one format into other formats so that messages from one application can be understood by other applications.

3. Message Routing: Messages should be routed based on rules rather than application logic. The routing mechanism should make it easy to quickly adapt to changing business requirements. For example, an application (sender) can send a message without any target information. The ESB should be able to route the message based on the message content. Hence, if a business requirement changes, the applications need not be modified, only the routing mechanism (rules) may have to be changed.

4. Transport Protocol Switching: The ESB should be able to transform messages so that messages received from the sender in one protocol can be sent to the receiver in another protocol. This transformation should support not only the Web services protocols (SOAP/HTTP), but other transport protocols (TCP/IP multicast, SCADA, etc).
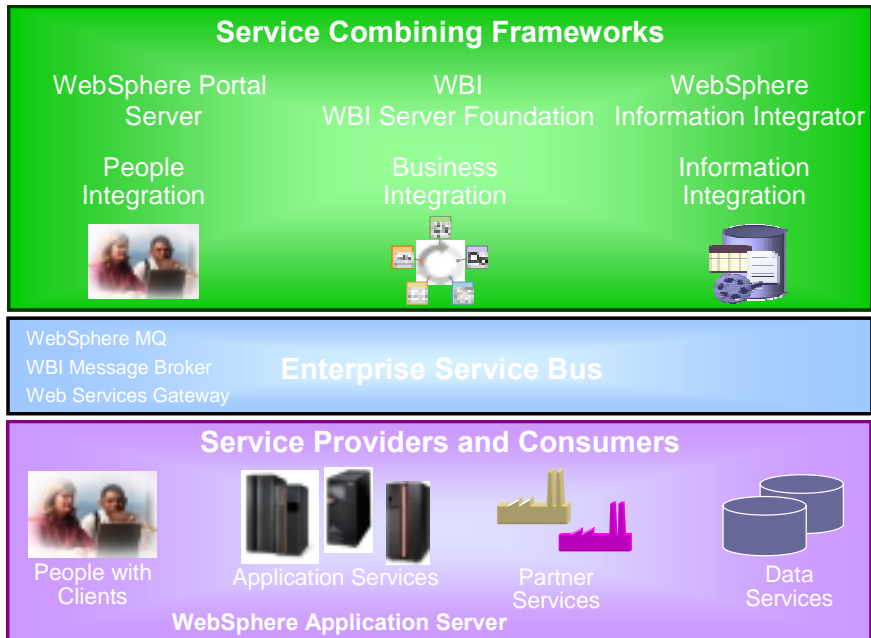
SOA Step 3 – Connect Services with the Enterprise Service Bus

Integrate Services Using Service Combining Frameworks

The final step in creating an SOA is to combine the various services through flexible service combining frameworks. Frameworks would handle the following tasks commonly found in enterprises:

1. People Integration: Portal and collaboration tools are needed for effective people integration. A portal presents an integrated interface through which users can interact with multiple applications. Services like e-mail, calendar, team rooms, e-learning, and e-meetings are essential for users to effectively collaborate.

2. Business Integration: In an SOA, a business process is made up of multiple components and is initiated by a request from a client. The business process may consume multiple services-–application services, partner services, data services. A service combining framework is needed to perform the choreography of the services—calling them in the proper order and passing and receiving the correct information from them. Each of the calls (asynchronous messaging, message transformation, message routing, etc.) by the process choreographer can be mediated by the enterprise service bus if required.

3. Information Integration: An important feature of an SOA infrastructure is the ability to seamlessly integrate and access heterogeneous data sources. Middleware providing information integration capabilities is necessary to provide a consistent view to different data sources. For example, a user who wants to access customer data (residing in DB2), policies (in SQL server), claims information (XML data), and claims payment data (in CRM/ERP systems) with one program request, should be able to write a single SQL query to access the heterogeneous data sources.

Service Combining Frameworks

WebSphere Portal Server   WBI WBI Server Foundation   WebSphere Information Integrator

People Integration   Business Integration   Information Integration

WebSphere MQ
WBI Message Broker
Web Services Gateway
Enterprise Service Bus

Service Providers and Consumers

People with Clients   Application Services   Partner Services   Data Services

WebSphere Application Server

Step 4 – Integrate Services Using Service Combining Frameworks

How Does IBM Compare to Microsoft and SAP?

IBM's Broader Vision for SOA

Many IT vendors have a narrow vision for building SOA solutions—using Web services alone to build SOA infrastructures. Hence, they require all services to be represented as Web services. IBM takes a much broader approach to SOA with a commitment to heterogeneous integration capabilities. IBM's vision for SOA is based on advanced developer tools to easily enable IT components as services, a powerful ESB with advanced mediation capabilities to interface with the various services, and *ready-made* service combining frameworks to jumpstart the SOA adoption by the enterprise.

IBM's WebSphere platform enables all IT components, including Web services, to be represented as reusable services. WebSphere provides over 58 technology and application adapters [8] for existing IT resources such as CICS, SAP®, and PeopleSoft® applications. These IT resources can be directly described in WSDL and represented as services on the ESB.

The approach of requiring a component to be represented as a Web service will often add performance overhead (in addition to the development effort required to convert it into a Web service). Alternatively, IBM's approach allows an application to choose the most appropriate and best performing method. For example, because invoking a service via SOAP requires considerable processing overhead for the parsing and formatting of XML messages, an internal client may choose to invoke a component using the faster RMI protocol rather than SOAP.

Business Components as Services

1. Building Web Services

√ IBM
IBM WebSphere Studio Application Developer (WSAD) [9] provides wizards to construct Web services out of Java classes (Java beans), Enterprise Java Beans (EJBs), process flows, Message Driven Beans (MDBs), DB2 Stored Procedures, DB2 XML Extender calls and SQL queries, and enterprise resources such as CICS, IMS, and SAP.

By using wizards, a programmer writes no code to accomplish these tasks. The programmer fills in dialog boxes and a wizard generates the code. The wizard generates a Web service, which is described in WSDL.

• Microsoft
Microsoft's SOA programming model is complicated by multiple object models and the need for manual programming. Unlike J2EE, which has one consistent object model, Microsoft's programming environment consists of three object models—.NET, COM, and COM+—each having different support for Web services. COM+ is the only Microsoft environment providing important services such as automatic transactions. COM objects and .NET objects need to rely on COM+ services to provide such functions. Creating a Web service for each of the object models is more complicated and time-consuming than in IBM's wizard-driven environment.

In the .NET framework, to create a Web service from a .NET object, a programmer first uses Microsoft Visual Studio® .NET to create a Web service skeleton. The programmer then has to write code to define the methods that the Web service will have, including the input and output data formats. These methods must instantiate the existing component, call the appropriate methods, and return the results to the original caller.

For the COM environment, Microsoft provides a SOAP Toolkit. To use the toolkit, the COM object must be contained in an ActiveX® DLL. Taking an existing COM object and placing it in an ActiveX DLL may require modification of the object. The SOAP Toolkit can expose COM objects contained in ActiveX DLLs as Web services but with limitations. The primary limitation is that currently the SOAP toolkit does not offer support for many of the Web service standards like WS-Security. In addition, with the SOAP toolkit several manual steps need to performed. Morever, the SOAP Toolkit is a stand-alone tool that is separate from both Visual Studio 6 and Visual Studio .NET.

To expose a COM+ 1.5 component as a secure, open-standards-based Web service, significant programming is required. To use the proprietary COM+ 1.5 security, an SSL certificate must first be installed. Such an application would not publish WSDL and would require the .NET Framework 1.1 on the client. Clients on other platforms, such as J2EE or .NET Framework 1.0 clients, can access the Web service only if COM+ 1.5 security is turned off. Microsoft provides no tooling to expose the earlier COM+ 1.0 components as Web services.

● **SAP NetWeaver™**
Programming SAP NetWeaver also involves mixed programming models—SAP's proprietary ABAP™ and J2EE. Different tools are needed—ABAP Workbench and NetWeaver Developer Studio—to develop Web services on these programming models. This can require managing multiple code repositories and a complex development environment.

NetWeaver provides wizards for Entity Beans and Session Beans. However there are no wizards for enabling databases or MDBs. In addition, exposing SAP process flows as Web services involves a lot more deployment steps than exposing IBM process flows as Web services.

2. Enabling Existing Enterprise Information Systems as Services

√ **IBM**
IBM WebSphere has numerous technology and application adapters (over 58) through which many enterprise systems can be easily represented as services (not necessarily Web services). IBM WSAD provides a common adapter framework with wizards to represent existing enterprise resources such as CICS, IMS, SAP, Siebel®, i2®, PeopleSoft, and Ariba® as services on the ESB. Because of the adapters, IBM WSAD wizards have the ability to "introspect," or examine, these resources and see which business objects are available. The programmer then uses the wizards to specify which business objects should be accessible through the service and what operations may be performed on them.

● **Microsoft**
The .NET environment is significantly more limited in its ability to connect to the spectrum of enterprise applications in use. Unlike IBM, Microsoft has no wizards with the ability to introspect enterprise applications and create services. To access CICS or IMS, Microsoft requires the use of a separate product called the Microsoft Host Integration Server and the associated Transaction Integrator (TI, formerly known as COMTI). For mySAP® Business Suite, Microsoft provides a BizTalk® adapter. However Microsoft has no adapters for PeopleSoft, Seibel, i2, Ariba, and other common enterprise applications. Creating a Web service from these resources requires extensive programming in the Microsoft environment.

● **SAP NetWeaver**
SAP is also limited in its ability to create services from the wide range of enterprise systems in use. SAP has no wizards to connect enterprise resources other than its own SAP applications. Without third-party adapters, enterprise resources have to be accessed using standard protocol adapters such as File, HTTP, SOAP, and JMS. Third-party adapters are needed to connect to systems like CICS, IMS, Siebel, etc.

3. Testing

√ **IBM**

IBM WebSphere Studio Application Developer (WSAD) is the only vendor in this group that provides an integrated test environment within its toolset. When a Web service is built, WSAD has the ability to automatically generate a sample application, or test client, which can invoke all of the methods of the Web service and display the results. Furthermore, the Web service can be run on the application server embedded in WSAD, so it not need be deployed to an external server for testing.

● **Microsoft**

In Microsoft Visual Studio .NET, a test client is not automatically generated. It must be programmed. Although Visual Studio .NET does allow a developer to launch Internet Explorer with a test page to access a Web service built in Visual Studio .NET, there are three shortcomings with this approach. First, the test of the Web service does not use the SOAP interface as it would use in production. Second, only simple data types can be tested, making the browser interface unable to work with many typical business applications. Third, this only works for Web services built by Visual Studio .NET. The IBM tool builds a test client based on a WSDL description and will work for any standards-based Web service.

● **SAP NetWeaver** has no automated testing tools.

4. Security

√ **IBM**

Building and deploying secure Web services is easy with WebSphere Application Server. Security is not specified within the WebSphere program itself, rather a declarative security model is used. At the time the Web service is deployed, security is specified in the deployment descriptor, which is an XML document external to the Web service program. This makes security easy to manage, change, and coordinate among multiple applications and Web services.

● **Microsoft**

In the Microsoft system, security is programmed into the Web service source code itself. Role-based access control in .NET is specified using metadata attributes in source code. This means that to change security, the source code must be modified, making it more difficult to manage, change, or coordinate with other applications or Web services.

Enterprise Service Bus

The main IBM products that comprise the ESB are WebSphere MQ (WMQ), WebSphere Business Integration Message Broker (WBIMB) and WebSphere Application Server [10]. IBM's ESB has native support for various transport protocols, and provides numerous adapters using which a wide range of technologies and applications can be connected to the ESB. Because of these adapters numerous legacy applications can be easily represented as services on the ESB. In addition, the ESB has tight integration with the WebSphere Application Server, allowing all applications to gain the advantages of the application server for Web services.

1. Connection

√ **IBM**

WebSphere MQ (WMQ) [11], IBM's messaging solution, connects to more than 35 platforms, including a wide variety of hardware and software from different vendors. This broad connectivity allows applications running on almost any hardware platform or operating system to exchange messages. It provides secure transactional message delivery. WebSphere MQ supports both point-to-point messaging and publish-and-

subscribe messaging. According to industry analysts, WebSphere MQ has over 80% market share [12]. It is currently used at more than 10,000 customer sites and more than 350 of IBM's top 500 customers.

WebSphere MQ Everyplace is a complementary messaging and queuing product, designed for smaller, lightweight, and wireless system environments like Palm OS, Pocket PC, and embedded systems. WebSphere MQ Everyplace provides a toolkit for building the MQe messaging environment and also provides a bridge between MQ Queue Manager and MQe environments. It has a tiny footprint; depending on the options selected, it can be as small as 100 KB. Special options can be enabled for wireless environments to accommodate varying bandwidth and encryption levels.

WebSphere MQ also can be used as the JMS messaging provider in WebSphere Application Server. This has some significant implications:

1. Any application connected to WebSphere MQ can be exposed as a service (including Web services), either through the WebSphere Business Integration Message Broker (WBIMB) [13] or WebSphere Application Server. This is also true of devices or applications connected by MQe or MQdsp (pervasive and telemetry). This is the power of IBM's ESB. Once an application is connected to the ESB, it can be exposed as a service and can access all the capabilities on the ESB.

2. All of the tooling to use the WBIMB is integrated in IBM WSAD so it can be used in an integrated way with the application server.

● **Microsoft**
Microsoft's messaging, MSMQ®, only connects Windows platforms to other Windows platforms. JMS messaging is not supported. Furthermore, MSMQ only supports point-to-point messaging and does not support publish-and-subscribe. In our internal tests we have found the performance of MSMQ to be significantly less than that of WMQ.

● **SAP NetWeaver**
SAP NetWeaver lacks a full-function messaging product like WebSphere MQ. Messaging in NetWeaver is supported only via an implementation of the Java Messaging Service (JMS). JMS runs in the application server, so using NetWeaver messaging requires the cost, footprint, and administration associated with a complete application server. Moreover, NetWeaver's application server Web AS is supported on significantly fewer platforms than is WebSphere MQ. Unlike a complete messaging product, there is no support for a network of servers, clustered queues, or centralized administration of multiple servers. NetWeaver's messaging component also does not support Web-scale publish/subscribe, pervasive, or remote sensing devices. In summary, to enable messaging across platforms, devices, and languages, a NetWeaver environment must use an enterprise messaging product like WebSphere MQ.

2. Message Transformation, Routing, and Transport Protocol Switching

√ **IBM**
IBM WebSphere Business Integration Message Broker (WBIMB) supports high-speed message transformation and routing. With this capability the format and destination of a message can be administered outside, rather than inside, an application. WBIMB receives a message, transforms it and determines its destination, then sends the message to the determined destination ("message routing"). This greatly facilitates application reuse because the application does not have to change to change the message format or message destination. It also simplifies management of the messaging network.

WBIMB has native support for seven transport protocols – WebSphere MQ Client for MQ Message Protocol, WebSphere JMS Client for MQ Message Protocol, WebSphere Web Services for SOAP/HTTP, WebSphere MQ Multicast for TCP/IP Multicast, WebSphere MQ Real-time for Internet Sockets, WebSphere MQE Client for MQ Mobile Message Protocol, and WebSphere MQ Telemetry for SCADA.

Thus, a message received in any protocol from the sender application, not just Web services protocol (SOAP/HTTP), can be transformed to the protocol supported by the receiver application. In addition to these protocols, WBIMB can send messages using a Web-scale publish-and-subscribe mechanism (large-scale distribution on the Internet via direct socket connection) or a multicast publish-and-subscribe mechanism. WebSphere also has 58 adapters to connect to a wide range of technologies and applications. Because of these adapters, numerous legacy applications can be easily represented as services on the ESB, *without* converting them into Web services.

The WBIMB tooling is integrated in IBM WSAD, and it implements the message flow paradigm, which is optimized for programming messages flows. The tooling has an integrated visual debugger that allows single stepping through the message flow.

● **Microsoft**
Microsoft does not have a high-speed message broker product like WBIMB. Although BizTalk can be programmed to perform some of WBIMB's capabilities like message routing and transformation, the process flow architecture and the use of the Message Box database for storing intermediate results means that BizTalk's performance can't achieve the same high levels that WBIMB can. Customers who value performance in an integration bus are encouraged to compare the throughput of WBIMB and BizTalk.

● **SAP NetWeaver**
SAP NetWeaver also does not have a high-speed message broker. XI adapters must convert messages to and from a proprietary XI protocol (SOAP over HTTPS) to be processed by the Integration Broker, so users may find performance to be significantly less than that of IBM's WBIMB.

NetWeaver's Exchange Infrastructure (XI) Integration Broker can perform only point-to-point routing and transformation. WBI, on the other hand, uses semantic brokering, which reduces the number and complexity of message maps by basing a transformation on a common message format. With point-to-point routing and transformation, application integration with XI will involve much more development configuration and many more maps. In addition, without a Common Business Object Model like WBI, XI cannot reuse easily pre-built processes and interfaces, has no data awareness in its adapters, and has no cross-referencing to associate records between systems.

XI's Integration Broker cannot perform error handling, compensation, complex mapping, merging and splitting of messaging, and transactions without using its Business Process Management (i.e., process choreography) component. WBI Message Broker supports these features natively without extensive coding or process choreography.

3. Web Services Gateway

√ **IBM**
Providing a Web Services Gateway is another advantage IBM has over Microsoft and SAP. Web services have moved from experimental, intranet-based applications to production applications that are accessed by customers and partners over the Internet. Enterprises that provide Web services over the Internet will benefit from a Web Services Gateway. A gateway can manage the access of a company's external customers and partners to its internal systems as well as the access of internal systems to the outside world [14]. The IBM Web Services Gateway is part of the WebSphere Application Server Network Deployment [15] product,

IBM's Web Services Gateway provides management functions, including access control, routing of requests, logging, and protocol conversion, The gateway acts as an intermediary, or broker, between the party requesting to use the Web service application and the party hosting or providing the Web service application. By acting as an intermediary, the Web Services Gateway provides flexibility and security to applications that employ Web services.

● **Microsoft** does not have a Web services Gateway.

● **SAP NetWeaver**
NetWeaver does not have a Web Services Gateway that offers out-of-the-box centralized administration of Web services, routing, protocol conversion, authorization, authentication, and audit. NetWeaver offers what it calls "enhanced Web services" in which its Exchange Infrastructure component is used to route and transform SOAP messages to other targets, message formats, and protocols. However, this technique requires considerable development effort for each Web service thus "enhanced." Moreover, whereas the purpose of a Web services gateway is to simplify the configuration and administration of a large number of services, NetWeaver's "enhanced Web services" also requires a great deal of developer effort to maintain as Web services are modified and redeployed.

Service Combining Frameworks

1. People Integration—Portal and Collaboration

√ IBM
The IBM WebSphere Portal Server (WPS) [16] is the leading product in the enterprise portal market space. Gartner positions WebSphere Portal in the leader quadrant [17, 18] and Butler group states WebSphere Portal to be the best of the available enterprise portal products [19]. WebSphere Portal consists of two offerings – Portal Enable, which is a base offering, and Portal Extend, which has powerful collaboration, search, and Web analysis features.

Portal Enable [20] delivers a single point of personalized interaction with applications, content, processes and people. Through business process integration, WebSphere Portal combines people and applications at a process level. The Portal's navigation paradigm is not only role-based, but also includes workflow orchestration that presents users with the tasks they need to complete, and all the information and applications needed to complete the task quickly. An integrated version of IBM Workplace™ Web Content Management [21] is also included with WebSphere Portal, which keeps the portal up-to-date with accurate content created by portal users.

Through the Portal Document Manager, users can easily share, view, and organize files of all types, ranging from documents to spreadsheets within the Portal community. The IBM DB2 Content Manager runtime repository is included to manage all forms of content (Web, e-mail, documents, digitized paper documents, images, audio/video, text messages) consistently. WebSphere Portal Application Integrator allows business users to quickly create portlets without any programming for interacting with relational databases, IBM Lotus® Domino® databases, and enterprise applications like SAP, Siebel, and PeopleSoft.

WebSphere Portal Extend [22], in addition to providing the basic functionality of Portal Enable, features the Collaboration Center—a set of ready-to-use collaborative portlets that can be used out of the box, providing instant value for portal users. The Collaboration Center integrates portlets for instant messaging, team workplaces, people finder, and e-meetings.

In the table below, several key portal capabilities are listed, with a brief comparison between portal products from IBM, Microsoft SharePoint, and SAP NetWeaver Enterprise Portal.

	IBM	Microsoft	SAP NetWeaver
Built-in Multi-Channel Support	Built-in support for browser, PDA, cell phone, voice.	No support.	Limited support for pervasive devices; no voice.
Internationalization	Support for both static and dynamic natural language translation via resource	No support for on-the-fly natural language translation.	No support for on-the-fly natural language translation.

	bundles and machine. translation, respectively.		
Integrated Content Management and Personalization	Industry-leading content management framework and rules/recommendation engine with ability to run campaigns.	Similar content management requires Microsoft Content Management Server. Personalization has no closed loop analysis.	Content management consists mostly of document management capability with limited ability to do personalization.
Process Choreography	WebSphere Portal Extend comes with the WebSphere BPEL container, providing the ability to deploy and integrate with business processes.	Customer must buy BizTalk Server for minimal flow support.	Must use NetWeaver XI for minimal flow support.
Integrated Collaboration	WebSphere Portal Extend comes with the IBM Instant Messaging server, allowing seamless integration of presence awareness and instant messaging.	Similar functionality requires purchase of Live Communications Server.	Limited presence awareness and reliance on third party products. SAP has no groupware product and integration with leading groupware products is limited to iFrames. SAP has no instant messaging capability that can be used across the enterprise for all users (portal and non-portal).
ISV Support and Developer Community	WebSphere Portal as a platform has the largest community of ISVs and developers building new functionality.	ISV and developer community limited to the Windows platform.	Very limited ISV and developer community.
Support for Portal Standards – JSR 168, JSR 170 and WSRP	WebSphere supports the three key portal standards.	No support for JSR 168 and JSR 170 standards.	NetWeaver cannot display WSRP remote portlets and there are no tools to produce JSR 168 and JSR 170 compliant portlets.
Application Integrator	WebSphere Portal Application Integrator can create portlets for many enterprise systems without programming.	No out-of-the-box application integrator for accessing non-Microsoft systems.	Does not have an out-of-the-box application integrator for integrating enterprise systems.

2. Business Integration – Process Choreography

√ IBM

IBM has a powerful, integrated product set for the entire choreography process -- modeling, deployment, and monitoring of business processes.

IBM WebSphere Business Integration (WBI) Modeler [23] is IBM's flagship product for business process modeling. Users can graphically design the way the processes will work, by both modeling current processes and planning future processes. Zero programming is involved in process modelling. The business process can include both human and automated steps. WBI Modeler can also simulate and analyze processes. Contingency analyses with "What if" scenarios can be run. Each task in the process can be assigned parameters like time and cost, and simulations can be run to improve the process model iteratively.

Once the process is modeled, deployment is fairly easy. The WBI Modeler generates a BPEL description. The BPEL description can be imported into IBM WSAD Integration Edition. The required services are then dragged and dropped in. Finally the entire process is deployed on the WebSphere Business Integration Server Foundation (WBISF) runtime [24].

WebSphere allows the administrator to view all running process instances (policy applications). The processes can be filtered along the following parameters:

- State (running, finished, failed, terminated, compensated).
- User who started the process.
- Process name.
- Date range.

In addition, individual process instances can be monitored, including the data that was originally input and the steps that have been completed (who completed the step and when started/completed). The tasks can be transferred to other users and the completed process instances can be saved for archival or further analysis.

● Microsoft

BizTalk Server 2004 is Microsoft's product for business integration. BizTalk cannot simulate business processes. Microsoft Visio®, which can be used by business analysts for process modeling, is a drawing tool only. It cannot perform analysis or simulation of business process cost and resources. Process monitoring is not integrated with modeling.

The need to involve human steps in an overall structured process is quite common. BizTalk does not provide any formal support for this. BizTalk Human Workflow Services (HWS) provides only some low-level building blocks oriented around unstructured, or ad-hoc, workflow. These building blocks have to be combined by the developer, along with many detailed manual steps and much coding, into an overall solution. The HWS capabilities do not include a workflow client for a user to display and interact with task assignment lists, or an administrative interface to monitor and administer the individual occurrences of workflow execution. For example, there are no functions for an administrator to determine where in the overall flow a given instance resides, or to reassign a task from one user to another if the first user was not able to finish an assigned task.

Developing an HWS solution involves many undocumented low-level steps and procedures – such as modifying internal InfoPath document template resources manually with Notepad or an equivalent. These manually modified resources would then be overwritten the next time the developer made a change to the InfoPath template itself – a maintenance nightmare.

The adapter framework for BizTalk does not have a robust, architected way for handling error conditions involving the remote application or environment. The adapter must reside on the BizTalk machine, so it has no way to be involved in error recovery of the remote system, which would greatly complicate the administrator's job of re-synchronizing BizTalk and the remote machine upon recovery from network or application outage.

BizTalk only supports one kind of service – Web Services. Access to other types of resources is accomplished by different techniques. To access CICS, for example, requires using Microsoft Host Integration Server (HIS) to define a .NET component that acts as a proxy to interact with a host-based CICS system. The BizTalk orchestration can then call that proxy component as it can call any other .NET component – but that's a .NET component call, not a service call. There is no WSDL file associated with the CICS system to document and control the CICS interaction. Another example is SAP. To access an SAP system, you install an SAP adapter and use wizards to generate schemas to represent the SAP document formats. There is no WSDL file associated with the SAP system to document and control the SAP interaction.

● SAP NetWeaver

SAP NetWeaver does not have any modeling, simulation, or business process monitoring capabilities. Modeling is done with a third-party tool – ARIS™ from IDS Scheer. While ARIS is well integrated with Solution Manager for documenting SAP business processes, Solution Manager processes are not executable, so changes to them require configuration and coding changes to the underlying SAP systems. Also, neither ARIS nor Solution Manager can export BPEL to XI, so top-down modeling is not possible.

Human workflow cannot be called directly from an XI business process, and SAP supports human workflow with a separate tool that can handle human activities only within an SAP environment.

3. Information Integration

√ IBM

IBM WebSphere Information Integrator (II) [25] provides federation capabilities across multiple (structured and unstructured) data sources and reduces the need for application programming. For example, an end-user client application may need to access the summary of a customer account. Using WebSphere II, this request can be made in a single SQL call. The SQL call is run against a federated database composed of DB2 (customer data), Oracle (past transactions), SQL Server (account data), and XML data (current loan applications).

At the heart of WebSphere II is a single, bidirectional interface that enables multiple disparate content sources to look and act as one system. It provides the ability to deliver access to underlying content and workflow systems, enable users to organize and work with content assets and workflow items as if managed in one system, adhere to the security of those systems, and add federation services such as metadata mapping, federated search, and single sign-on.

WebSphere II also provides out-of-the-box connectors to leading content repositories to quickly unify a broad range of content sources and workflow systems without the cost, complexity, and risk of custom programming efforts. Connectors are available for the following applications, databases, and content management systems; federated searches can be performed across all these systems:

- File systems, HTTP/HTTPS, News groups (NNTP), text documents
- DB2 UDB, DB2 Content Manager, Informix®
- Oracle databases, Sybase®
- WebSphere MQ Workflow
- Lotus® Notes®, Lotus Domino.Doc®
- Documentum®
- FileNet® Content Services, FileNet Image Services, FileNet Image Services Resource Adapter, FileNet P8 Content Manager, FileNet P8 Business Process Manager
- Open Text Livelink™
- Microsoft Index Server/NTFS, SQL Server, Microsoft Exchange public folders
- Stellent Content Server™
- Interwoven™ TeamSite™
- Hummingbird™ Enterprise DM

● Microsoft

Microsoft SQL Server can link to multiple other ODBC compliant databases—SQL Server, Oracle, Access. Linkage is done through OLE DB providers. OLE DB is the previous generation of Microsoft database connection technology, and is based on COM. The newer connection technology used by Microsoft is ADO.NET. Shortcomings in OLE DB suggest the following IBM advantages:

- Better query optimization.
- Better caching.
- Federation with messaging (WebSphere MQ).
- Federation with a larger number of enterprise resources.

● SAP NetWeaver

SAP NetWeaver supports DB2, Oracle, SQL Server, and MaxDB as the back-end databases. However, it does not provide any federation capabilities.

Summary Scorecard

	IBM	Microsoft	SAP NetWeaver
Service Enablement	**+**	**?**	**?**
Building a New Web Service	WebSphere Studio provides wizards.	Multiple programming environments -- .NET, COM, COM+. Programming required.	Wizards only for Entity Beans, Session Beans. Databases, MDBs, etc. require programming.
Using Existing IT Components	Over 58 adapters to enable varied technologies and applications as services. No need to convert to Web services. Wizards create the code.	Application adapter only for mySAP. Other IT resources must first be converted into Web services; programming required.	Adapter only for SAP. Other IT resources must first be converted into Web services; programming required.
Testing	Integrated testing environment in WSAD.	No automated testing tools in the developer environment.	No automated testing tools in the developer environment.
Security	Declarative security model.	Role-based access control in .NET specified using metadata attributes in source code	Declarative security model.
ESB	**+**	**?**	**?**
Messaging	Fast messaging system with WebSphere MQ, industry-leading messaging platform.	Works only for connecting Windows systems. MSMQ is considerably slow. BizTalk has an adapter for WMQ.	Support only for limited number of platforms. No support for network of servers or clustered queues.
Message Routing and Transformation	High-performance message broker – WBIMB.	No high-performance broker. Routing and Transformation has to be done by BizTalk, which significantly decreases performance.	No high-performance broker.
Web Services Gateway	WebSphere Application Server Network Deployment	No product	No Product
Service Combining Frameworks	**+**	**?**	**?**
Portal and Collaboration	Powerful built-in features for collaboration, process choreography, content management. Portal standards are supported.	Features like collaboration, choreography, content management require additional Microsoft products. JSR portal standards are not supported	Limited collaboration, content management , choreography capabilities. JSR portal standards not supported.
Process Choreography/ Business Integration	Advanced business process modeling and simulation capabilities provided by WBI Modeler. Advanced human workflow capabilities.	Human workflow capabilities are limited in BizTalk. No simulation or analysis of business process models. Process monitoring not integrated with modeling	No modeling or monitoring capabilities. Third-party tools required.
Information Integration	Advanced federation capabilities. Can integrate data from standard databases, file systems, content management systems, application data, etc.	Limited federation capabilities. Can connect only to standard ODBC data sources.	No federation capabilities.

Conclusion

In this paper we have provided a detailed feature-by-feature comparison of IBM, Microsoft, and SAP NetWeaver products in their ability to implement, deploy, and monitor SOA applications. As our study shows, IBM provides a much broader and functionally rich platform for implementing an SOA infrastructure than does Microsoft and SAP.

Neither Microsoft nor SAP toolsets can match the ability of IBM's development tools to create Web services and drag-and-drop workflows. A number of developer productivity features unique to IBM's toolset, such as a built-in Web service test client, are missing in Microsoft's and SAP's tools. Microsoft and SAP NetWeaver development environments also suffer from the problems of multiple programming models and the lack of wizards to automatically convert existing enterprise IT applications into services.

Elements such as a sophisticated ESB with multi-protocol routing, message transformation, and a Web Services Gateway are missing in Microsoft's and SAP's portfolios, as are the advanced information integration and business integration tools and capabilities found in IBM's portfolio. Also missing in the Microsoft and SAP platforms are IBM's modeling, monitoring, and management capabilities for business process choreography.

Microsoft and SAP also fall short in comparison with IBM's service combining frameworks—WebSphere Portal Server for people integration, WBISF and WAS for process choreography, and WebSphere II for information integration—all of which comprise superior SOA frameworks in their respective areas than do Microsoft's and SAP's platforms.

Businesses are rapidly adopting SOA as a means to make their IT infrastructures more efficient, reliable, resilient, and adaptable to change. Many IT vendors have come up with various products to design and build SOA infrastructures. We invite you to make your own comparison of IBM's SOA capabilities vs. those of our competitors. We believe you will find that, in all the key areas of SOA infrastructure, IBM has the market-leading products and the deepest set of features and capabilities.

References

1. Elements of Service-Oriented Analysis and Design. Olaf Zimmerman, Pal Krogdahl, Clive Gee. http://www-106.ibm.com/developerworks/webservices/library/ws-soad1/
2. Service-oriented modeling and architecture. Ali Arsanjani. http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/
3. Webservices.org http://www.webservices.org
4. BPEL for Web Services http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
5. Web Services Description Language http://www.w3.org/TR/wsdl
6. Simple Object Access Protocol http://www.w3.org/TR/soap/
7. IBM WebSphere Software http://www-306.ibm.com/software/websphere/
8. WebSphere Business Integration Adapters http://www-306.ibm.com/software/integration/wbiadapters/
9. WebSphere Studio Application Developer http://www-306.ibm.com/software/awdtools/studioappdev/
10. Building an Enterprise Service Bus with Websphere Application Server, Parts 1 & 2. Rachel Reinitz, Andre Tost. http://www-128.ibm.com/developerworks/websphere/techjournal/0501_reinitz/0501_reinitz.html http://www-128.ibm.com/developerworks/websphere/techjournal/0502_reinitz/0502_reinitz.html
11. WebSphere MQ http://www-306.ibm.com/software/integration/wmq/
12. Mission Critical Messaging Middleware Market Description. Wintergreen Research 2005 http://wintergreenresearch.com
13. WebSphere Business Integration Message Broker http://www-306.ibm.com/software/integration/wbimessagebroker/
14. An Introduction to Web Services Gateway. Chandra Venkatapathy, Simon Holdsworth. http://www-128.ibm.com/developerworks/webservices/library/ws-gateway/
15. WebSphere Application Server ND http://www-306.ibm.com/software/webservers/appserv/was/network/
16. WebSphere Portal http://www-306.ibm.com/software/genservers/portal/
17. Gartner Dataquest identifies WebSphere Portal as the leader in enterprise portal market in 2003 with 21.9% market share. May, 2004
18. Gartner Positions IBM's WebSphere Portal in Leader Quadrant. Phifer, et al., Gartner, March 2004
19. An analysis of Enterprise Portal products. Mike Davis. Butler Group May 2003. ftp://ftp.software.ibm.com/software/websphere/pdf/Butler-WebSpherePortal.pdf
20. WebSphere Portal Enable http://www.ibm.com/software/webservers/portal/enable
21. IBM Workplace Web Content Management http://www.lotus.com/products/product5.nsf/wdocs/homepage
22. WebSphere Portal Extend http://www.ibm.com/software/webservers/portal/extend
23. WBI Modeler http://www-306.ibm.com/software/integration/wbimodeler/
24. WBI Server Foundation http://www-306.ibm.com/software/integration/wbisf/
25. IBM WebSphere Information Integrator http://www-306.ibm.com/software/data/integration/
26. "Web services Platform Shootout" by Forrester Research ftp://ftp.software.ibm.com/software/websphere/webservices/webservicesplatformshootout-oct.pdf
27. Web services 2002 -- Market Milestone Report by Delphi Group ftp://ftp.software.ibm.com/software/websphere/webservices/delphiws_wp_ibm.pdf
28. IBM WebSphere MQ Royal Academy of Engineering's MacRobert prize for technological and engineering innovation http://www-306.ibm.com/software/integration/wmq/raeaward.html